WHAT IS CLAIMED IS:

1. A method of adding a plurality of partial products, wherein each partial product has a plurality of bits having respective binary weights, wherein each bit can have a first or second logic state, the method comprising:

forming a first set of multiple-bit columns from bits of the plurality of partial products, wherein the bits in each column of the first set have the same binary weight; and

encoding each multiple-bit column in the first set into a respective modified partial product, which represents a number of bits in the column having the first logic state.

2. The method of claim 1 and further comprising:

forming a second set of multiple-bit columns from bits of the modified partial products, wherein the bits bit in each column of the second set have the same binary weight as one another; and

encoding each column in the second set into a respective further modified partial product, which represents a number of bits in the column having the first logic state.

3. The method of claim 1 and further comprising:

repeating the steps of forming and encoding to form successive sets of further modified partial products until the number of

further modified partial products is reduced to a desired number.

5. The method of claim 3 and further comprising:

adding the desired number of further modified partial products to produce a product.

6. The method of claim 1 wherein encoding comprises:

for each multiple-bit column in the first set, forming a modified column word by packing all the bits in that column having the first logic state into a first contiguous set of bit positions in the modified column word; and

generating the respective modified partial product based on a pattern of the first and second logic states in the modified column word.

7. The method of claim 6 wherein:

forming a modified column word comprises packing all the bits in that column having the first logic state to one end of the modified column word; and

generating comprises generating the respective modified partial product based on the bit positions in the modified column word at which values of the bits in the modified

column word transition between the first and second logic states.

8. The method of claim 6 wherein the modified column word comprises first and second opposite ends and the step of forming the modified column word comprises, for each multiple-bit column in the first set:

(a) grouping adjacent ones of the bits of the multiple-bit column into a present level of multiple-bit groups;

(b) for each group in the present level, packing any of the bits in that group having the first logic state toward a first end of that group, which corresponds in orientation to the first end of the modified column word, to form a respective modified present level group;

(c) grouping adjacent ones of the modified present level groups into respective higher level groups, with the higher level groups becoming the present level groups;

(d) repeating step (b) for the present level groups generated in step (c); and

(e) repeating steps (b) through (d) until there is only one present level group, which has the same number of bit positions as the modified column word and any of the bits in that group having the first logic state

have been packed toward the first end of that group.

9. The method of claim 8 wherein step (b) comprises shifting the bits in that group having the first logic state to different bit positions in that group through a multiplexer.

10. A method of multiplying a multiplicand by a multiplier, the method comprising:

generating a plurality of partial products, wherein each partial product has a plurality of bits having respective binary weights, wherein each bit can have a first or second logic state;

forming a first set of multiple-bit columns from bits of the plurality of partial products, wherein the bits in each column of the first set have the same binary weight; and

encoding each multiple-bit column in the first set into a respective modified partial product, which represents a number of bits in the column having the first logic state.

11. The method of claim 10 and further comprising:

forming a second set of multiple-bit columns from bits of the modified partial products, wherein the bits bit in each column of the second set have the same binary weight as one another; and

encoding each column in the second set into a respective further modified partial product, which represents a number of bits in the column having the first logic state.

12. The method of claim 11 and further comprising:

repeating the steps of forming and encoding to form successive sets of further modified partial products until the number of further modified partial products is reduced to a desired number.

13. The method of claim 12 and further comprising:

adding the desired number of further modified partial products to produce a product.

14. The method of claim 10 wherein encoding comprises:

for each multiple-bit column in the first set, forming a modified column word by packing all the bits in that column having the first logic state into a first contiguous set of bit positions in the modified column word; and

generating the respective modified partial product based on a pattern of the first and second logic states in the modified column word.

15. The method of claim 14 wherein:

forming a modified column word comprises packing all the bits in that column having the first logic state to one end of the modified column word; and

generating comprises generating the respective modified partial product based on the bit positions in the modified column word at which values of the bits in the modified column word transition between the first and second logic states.

16. The method of claim 14 wherein the modified column word comprises first and second opposite ends and the step of forming the modified column word comprises, for each multiple-bit column in the first set:

(a) grouping adjacent ones of the bits of the multiple-bit column into a present level of multiple-bit groups;

(b) for each group in the present level, packing any of the bits in that group having the first logic state toward a first end of that group, which corresponds in orientation to the first end of the modified column word, to form a respective modified present level group;

(c) grouping adjacent ones of the modified present level groups into respective higher level groups, with the higher level groups becoming the present level groups;

(d) repeating step (b) for the present level groups generated in step (c); and

(e) repeating steps (b) through (d) until there is only one present level group, which has the same number of bit positions as the modified column word and any of the bits in that group having the first logic state have been packed toward the first end of that group.

17. The method of claim 16 wherein step (b) comprises shifting the bits in that group having the first logic state to different bit positions in that group through a multiplexer.

18. The method of claim 10 wherein generating a plurality of partial products comprises, for each partial product:

multiplexing the multiplicand multiplied by zero, the multiplicand multiplied by one, the multiplicand multiplied by minus one, the multiplicand multiplied by two, and the multiplicand multiplied by minus two to produce the respective partial product.

19. A multiplier circuit comprising:

a partial products generator comprising a multiplicand input, a multiplier input and a plurality of partial product outputs, wherein each output has a plurality of bits

having respective binary weights and which can have a first or second logic state; and

adding means for forming a first set of multiple-bit columns from bits of the plurality of partial product outputs, wherein the bits in each column of the first set have the same binary weight, and encoding each column in the first set into a respective modified partial product, which represents a number of bits in the column having the first logic state.

20. The multiplier circuit of claim 19 wherein the partial products generator comprises a plurality of multiplexers, wherein each multiplexer multiplexes the multiplicand multiplied by zero, the multiplicand multiplied by one, the multiplicand multiplied by minus one, the multiplicand multiplied by two, and the multiplicand multiplied by minus two to a respective one of the partial product outputs.

21. The multiplier circuit of claim 19 wherein the adding means comprises:

means for forming a second set of multiple-bit columns from bits of the modified partial products, wherein the bits bit in each column of the second set have the same binary weight as one another; and

encoding means for encoding each column in the second set into a respective further

modified partial product, which represents a number of bits in the column having the first logic state.

22. The multiplier circuit of claim 21 wherein the adder means further comprises:

means for repeating the steps of forming and encoding to form successive sets of further modified partial products until the number of further modified partial products is reduced to a desired number.

23. The multiplier circuit of claim 19 wherein the adding means comprises:

packing means for each multiple-bit column in the first set, for forming a modified column word by packing all the bits in that column having the first logic state into a first contiguous set of bit positions in the modified column word; and

generating means for generating the respective modified partial product based on a pattern of the first and second logic states in the modified column word.

24. The multiplier circuit of claim 23 wherein:

the packing means comprises means for packing all the bits in that column having the first logic state to one end of the modified column word; and

the generating means comprises means for generating the respective modified partial product based on the bit positions in the modified column word at which values of the bits in the modified column word transition between the first and second logic states.